



ARL-TR-7948 • FEB 2017



Generalized Optimal-State-Constraint Extended Kalman Filter (OSC-EKF)

by James M Maley, Kevin Eckenhoff, and Guoquan Huang

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Generalized Optimal-State-Constraint Extended Kalman Filter (OSC-EKF)

by James M Maley

Weapons and Materials Research Directorate, ARL

Kevin Eckenhoff and Guoquan Huang

Department of Mechanical Engineering, University of Delaware, Newark, DE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) February 2017		2. REPORT TYPE Technical Report		3. DATES COVERED (From - To) December 2015–May 2016	
4. TITLE AND SUBTITLE Generalized Optimal-State-Constraint Extended Kalman Filter (OSC-EKF)				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) James M Maley, Kevin Eckenhoff, and Guoquan Huang				5d. PROJECT NUMBER AH80	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-WML-F Aberdeen Proving Ground, MD 21005-5066				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-7948	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES primary author's email: <james.m.maley2.civ@mail.mil>.					
14. ABSTRACT Cameras, inertial measurement units (IMUs), and computational power have all become practical, and the demand for small and efficient navigation systems that don't rely on external infrastructure such as GPS is high. Combining visual information with inertial sensing is a challenging problem. The optimal-state-constraint extended Kalman filter (OSC-EKF) is a new method previously designed to optimally combine relative pose constraints from a monocular camera with the output of an IMU. This framework is generalized so that any combination of sensors that can be combined to produce relative pose constraints can be used to update the EKF. A stereo vision-structure and motion (SAM) problem and a monocular SAM problem are both used to update the OSC-EKF without making any changes to the EKF. The efficacy of these algorithms is demonstrated by achieving reasonable consistency and accuracy on a challenging micro aerial vehicle dataset.					
15. SUBJECT TERMS simultaneous localization and mapping, visual odometry, Kalman filter, positioning, navigation					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 34	19a. NAME OF RESPONSIBLE PERSON James M Maley
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 410-306-0814

Contents

List of Figures	iv
Acknowledgments	v
1. Introduction	1
2. State Definition	3
3. Propagation	5
4. OSC-EKF Update	8
5. Practical MLE Issues	12
6. Experimental Results	13
7. Conclusions and Future Work	16
8. References	17
Appendix A. Feature Parameterization	21
List of Symbols, Abbreviations, and Acronyms	24
Distribution List	25

List of Figures

Fig. 1	Depiction of OSC-EKF window structure	4
Fig. 2	Stereo position tracking (MH 05 difficult left, V2 03 difficult right)	14
Fig. 3	Position error for the MH 05 difficult sequence (stereo left, mono right)	14
Fig. 4	Position error for the V2 03 difficult sequence (stereo left, mono right).	15
Fig. 5	Consistency analysis for the MH 05 difficult sequence (stereo left, mono right)	15
Fig. 6	Consistency analysis for the V2 03 difficult sequence (stereo left, mono right)	16

Acknowledgments

The authors would like to thank Dr Frank Fresconi for performing a detailed technical review of this report.

INTENTIONALLY LEFT BLANK.

1. Introduction

Knowing the location and attitude of a mobile sensing platform is a prerequisite for many of the functions performed by robots, unmanned aerial vehicles, wearable devices, game controllers, and many other platforms. In many common applications, GPS and similar systems greatly simplify this problem by providing location and velocity information directly. However, GPS cannot operate indoors, underground, underwater, near GPS jammers, or on other planets. These and other factors can deny GPS availability, so there has been significant motivation in recent decades to create navigation solutions that do not depend on GPS.

One configuration that has received considerable attention is an inertial measurement unit (IMU) aided by landmark sightings and other visual information, both of which existed long before GPS. Inertial navigation has become prevalent due to recent advances in microelectromechanical system technology, which have resulted in the creation of chip-scale accelerometers and gyroscopes. These devices have very low size, weight, and power (SWaP) requirements and can be integrated at a very high rate to provide relative position, velocity, and attitude. The problem is that even under ideal circumstances the state estimates from an IMU diverge as a function of time due to noise. The drift is much greater if there are deterministic errors in any of the sensor outputs. At the same time, extremely low SWaP cameras have become ubiquitous, and a host of computer vision algorithms have been invented to identify points of interest in an image, and track them reliably from frame to frame.¹⁻³ Utilizing these correspondences, it is possible to aid the IMU with relative rotation and translation information that drifts as a function of what the camera can see, rather than time. An IMU aided with computer vision data for the purpose of navigation is known as visual-inertial navigation system (VINS). Purpose-built VINS platforms exist,⁴ and many others have been constructed with low-cost components.

Visual-inertial simultaneous localization and mapping (SLAM) and visual-inertial odometry (VIO) are 2 closely related algorithm categories that are used to blend image and IMU information into a navigation solution. The purpose of SLAM⁵⁻⁷ is to estimate the locations of the observed landmarks and the pose of the vehicle relative to the map, whereas VIO uses the landmarks, but the primary output is only vehicle pose because it is typically assumed that there will be no opportunity for loop closure. Both problems are challenging because the visual measurements

are nonlinear, tightly coupled with the vehicle pose, and usually great in number. Despite the heavy computational cost of these algorithms, multiple real-time implementations have been demonstrated.^{3,8–11} However, there is still significant room for improvement in computational complexity, accuracy, and robustness in these algorithms.

Early work in visual odometry is reviewed succinctly in Scaramuzza and Fraundorfer¹² and Fraundorfer and Scaramuzza.¹³ Most VIO algorithms proposed in the last decade use either a Kalman filtering approach, or a sliding-window batch-optimization approach to fuse visual correspondences with IMU output. Batch-optimization approaches require a history of past poses and landmark correspondences, and adjust vehicle poses, landmark locations, and usually sensor errors to achieve a maximum likelihood estimate (MLE) of these parameters. With visual data, this is typically referred to as bundle adjustment or structure and motion (SAM).^{14,15} This problem can quickly grow to unmanageable size for real-time implementation. This issue can be addressed by using keyframes¹⁶ and solving the MLE incrementally as measurements and poses are accumulated.^{17,18} “Structure-less” methods seek to reduce complexity by eliminating the need to estimate landmark locations. This has been achieved algebraically by using 3-view constraints¹⁹ and by projecting the residuals onto the null-space of the sensitivity matrix corresponding to landmark location errors.²⁰ The efficiency of using IMU measurements in a SAM framework has been improved by preintegration techniques.^{20,21}

Most recent filter-based methods are based off of the structure of the multistate constraint Kalman filter (MSCKF),²² in which the strapdown navigation error-states are augmented by a sliding window of “cloned” pose error states.²³ The landmark locations are triangulated from the poses in the sliding window, and the correlation between state and landmark location errors is avoided in the EKF update by projecting the residuals onto the null-space of the sensitivity matrix corresponding to landmark location errors (i.e., the residuals are no longer sensitive to the landmark location errors). This approach has become very popular because of its efficiency, and many improvements to the basic structure have since been made. The consistency of the estimator has been improved by enforcing the semi-group constraint on propagation Jacobian²⁴ and enforcing constraints that prevent the filter from gaining information in unobservable directions^{25–27}

The optimal-state-constraint extended Kalman filter (OSC-EKF) presented in²⁸ utilizes an image-only SAM process over a fixed window of previous and current states to estimate the relative camera poses. This SAM problem is much smaller than what would be solved using a sliding window batch-optimization technique, and can be solved efficiently due to its sparse nature using various open-source solvers. The structure is marginalized out, and the motion data is combined with the IMU predictions using an EKF. The complexity of the dense matrix computations is therefore only dependent on the number of cloned poses, which is user-selectable. The prior work was presented specifically for an IMU aided with a monocular camera. The novel contribution presented here is a generalized version of the OSC-EKF, that allows for the IMU to be aided by any measurements that can produce relative pose constraints based on a MLE. To demonstrate this point, the algorithm is used on the European Robotics Challenge (EuRoC) micro aerial vehicle datasets²⁹ using both monocular and stereo configurations, without making any changes to the OSC-EKF. Instead, only implementing the SAM required modification.

2. State Definition

The OSC-EKF algorithm works by processing the measurements from several image frames at once at every EKF update, as shown in Fig. 1. The time distance between local frames (L_0, L_1 , etc.) is user defined and in this work is chosen to be 10 camera frames. Between consecutive local frames (e.g., L_0 to L_1), the inertial measurement unit is integrated to produce an “IMU trajectory” (i.e., the predicted state of the IMU) at the intermediate image frames (I_1, I_2 , etc.). Concurrently, feature points are tracked between the image frames and when the next local frame is reached, a SAM problem is solved to estimate the IMU state at the intermediate image frames, and the 3-D feature point locations and other nuisance parameters relative to the local frame (i.e., the SAM trajectory). Other relative measurements such as wheel odometry or light detection and ranging (LIDAR) measurements could be used as well. The OSC-EKF update step then optimally combines the IMU and SAM trajectories, which will be described in more detail in the following sections.

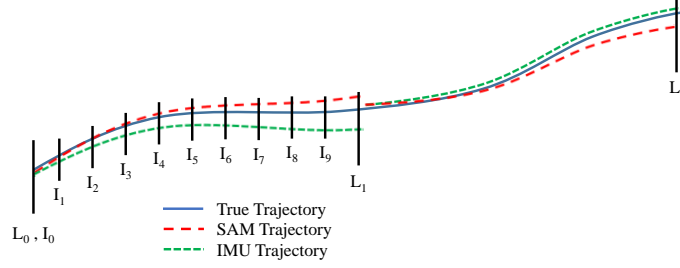


Fig. 1 Depiction of OSC-EKF window structure

The state vector is identical to other sliding-window EKFs such as the MSCKF, in that it consists of the current IMU quaternion, velocity, position (${}^{I_k}\bar{\mathbf{q}}$, ${}^G\mathbf{v}_{I_k}$ and ${}^G\mathbf{p}_{I_k}$, respectively), the gyroscope, and accelerometer biases (\mathbf{b}_{g_k} and \mathbf{b}_{a_k}), and the previous l poses (quaternions and positions). The total state vector relative to the global reference frame is

$${}^G\mathbf{x} = \left\{ {}^G\mathbf{x}_1^T \quad {}^G\mathbf{x}_2^T \right\}^T. \quad (1)$$

The IMU states are contained in ${}^G\mathbf{x}_1$, while the sliding window of the previous l IMU poses are contained in ${}^G\mathbf{x}_2$:

$${}^G\mathbf{x}_1 = \left\{ {}^{I_k}\bar{\mathbf{q}}^T \quad \mathbf{b}_{g_k}^T \quad {}^G\mathbf{v}_{I_k}^T \quad \mathbf{b}_{a_k}^T \quad {}^G\mathbf{p}_{I_k}^T \right\}^T, \quad (2)$$

$${}^G\mathbf{x}_2 = \left\{ {}^{I_{k-1}}\bar{\mathbf{q}}^T \quad {}^G\mathbf{p}_{I_{k-1}}^T \quad \dots \quad {}^{I_{k-l}}\bar{\mathbf{q}}^T \quad {}^G\mathbf{p}_{I_{k-l}}^T \right\}^T. \quad (3)$$

The OSC-EKF utilizes an indirect error-state model,³⁰ the filter estimates error-states, which are used to update the estimated states external to the filter itself. The error-states (denoted by $\tilde{\bullet}$) are defined by how they are applied to the estimated states (denoted $\hat{\bullet}$) to obtain the true states. For the biases, positions, and velocities, the error states are additive, and defined as follows:

$$\begin{aligned} {}^G\mathbf{p}_{I_k} &= {}^G\tilde{\mathbf{p}}_{I_k} + {}^G\hat{\mathbf{p}}_{I_k}, \\ {}^G\mathbf{v}_{I_k} &= {}^G\tilde{\mathbf{v}}_{I_k} + {}^G\hat{\mathbf{v}}_{I_k}, \\ \mathbf{b}_{g_k} &= \tilde{\mathbf{b}}_{g_k} + \hat{\mathbf{b}}_{g_k}, \\ \mathbf{b}_{a_k} &= \tilde{\mathbf{b}}_{a_k} + \hat{\mathbf{b}}_{a_k}. \end{aligned} \quad (4)$$

The quaternions in this work use a multiplicative error-state model, in which the true quaternion is represented by the estimated quaternion, followed by an additional

rotation through an error quaternion:

$${}^G \bar{\mathbf{q}} = {}^G \tilde{\mathbf{q}} \otimes {}^G \hat{\mathbf{q}}, \quad (5)$$

where \otimes is the quaternion multiplication operator. The estimated rotation is assumed to be close enough to the true rotation that the error quaternion can be approximated by a vector of small rotation angles ${}^G \tilde{\boldsymbol{\theta}}$, which enables the use of the following approximations:

$${}^G \tilde{\mathbf{q}} \approx \begin{Bmatrix} {}^G \tilde{\boldsymbol{\theta}} \frac{1}{2} \\ 1 \end{Bmatrix}, \quad (6)$$

$$\mathbf{C} \left({}^G \bar{\mathbf{q}} \right) \approx \left(\mathbf{I}_3 - \left[{}^G \tilde{\boldsymbol{\theta}} \times \right] \right) \mathbf{C} \left({}^G \hat{\mathbf{q}} \right). \quad (7)$$

The error-state vector that is estimated by the OSC-EKF is

$${}^G \tilde{\mathbf{x}} = \left\{ {}^G \tilde{\mathbf{x}}_1^T \quad {}^G \tilde{\mathbf{x}}_2^T \right\}^T. \quad (8)$$

Where

$${}^G \tilde{\mathbf{x}}_1 = \left\{ {}^G \tilde{\boldsymbol{\theta}}^T \quad \tilde{\mathbf{b}}_{g_k}^T \quad {}^G \tilde{\mathbf{v}}_{I_k}^T \quad \tilde{\mathbf{b}}_{a_k}^T \quad {}^G \tilde{\mathbf{p}}_{I_k}^T \right\}^T, \quad (9)$$

$${}^G \tilde{\mathbf{x}}_2 = \left\{ {}^{G_{k-1}} \tilde{\boldsymbol{\theta}}^T \quad {}^{G_{k-1}} \tilde{\mathbf{p}}_{I_{k-1}}^T \quad \dots \quad {}^{G_{k-l}} \tilde{\boldsymbol{\theta}}^T \quad {}^{G_{k-l}} \tilde{\mathbf{p}}_{I_{k-l}}^T \right\}^T. \quad (10)$$

3. Propagation

Like all discrete-time Kalman filters, the OSC-EKF requires a method of predicting the current state's mean and covariance from the mean and covariance at the previous time step. The true IMU states are governed by continuous-time kinematic relationships, which are used to propagate the estimated IMU states, and linearized to propagate the IMU error state covariance. The sliding window of previous poses is managed by a process that has come to be called “stochastic cloning”.²³ The true IMU states propagate according to the following continuous-domain kinematic relationships:

$$\begin{aligned} {}^G \dot{\bar{\mathbf{q}}} &= \boldsymbol{\Omega} \left({}^G \boldsymbol{\omega} \right) {}^G \bar{\mathbf{q}}, \\ {}^G \dot{\mathbf{p}}_I &= {}^G \mathbf{v}_I, \\ {}^G \dot{\mathbf{v}}_I &= {}^G \mathbf{a}_I, \\ \dot{\mathbf{b}}_g &= \mathbf{n}_{wg}, \\ \dot{\mathbf{b}}_a &= \mathbf{n}_{wa}, \end{aligned} \quad (11)$$

where ${}^I\boldsymbol{\omega}$ is the angular velocity of the IMU with respect to the global coordinates, viewed in IMU coordinates, ${}^G\mathbf{a}_I$ is the total acceleration of the IMU, \mathbf{n}_{wg} and \mathbf{n}_{wa} are white noise processes that drive the IMU biases, and the matrix $\boldsymbol{\Omega}({}^I\boldsymbol{\omega})$ is defined as follows:

$$\boldsymbol{\Omega}({}^I\boldsymbol{\omega}) = \begin{bmatrix} -[{}^I\boldsymbol{\omega} \times] & {}^I\boldsymbol{\omega} \\ -{}^I\boldsymbol{\omega}^T & 1 \end{bmatrix}. \quad (12)$$

The IMU is assumed to provide an angular velocity measurement ${}^I\boldsymbol{\omega}_m$ from the gyroscope and a specific force measurement ${}^I\mathbf{a}_m$ from the accelerometer. The models for these measurements are given by

$${}^I\boldsymbol{\omega}_m = {}^I\boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g, \quad (13)$$

$${}^I\mathbf{a}_m = \mathbf{C}({}^I\hat{\mathbf{q}})({}^G\mathbf{a}_I - {}^G\mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a, \quad (14)$$

where ${}^G\mathbf{g}$ is the gravitational acceleration expressed in the global frame, and \mathbf{n}_g and \mathbf{n}_a are white noise processes. The estimated states are propagated forward in time by integrating the kinematic equations using the IMU measurements and assuming the error states and noise are zero:

$$\begin{aligned} {}^I_G\dot{\hat{\mathbf{q}}} &= \boldsymbol{\Omega}({}^I\hat{\boldsymbol{\omega}}){}^I_G\hat{\mathbf{q}}, \\ {}^G\dot{\hat{\mathbf{p}}}_I &= {}^G\hat{\mathbf{v}}_I, \\ {}^G\dot{\hat{\mathbf{v}}}_I &= {}^G\hat{\mathbf{a}}_I, \\ \dot{\hat{\mathbf{b}}}_g &= \mathbf{0}, \\ \dot{\hat{\mathbf{b}}}_a &= \mathbf{0}, \end{aligned} \quad (15)$$

where ${}^I\hat{\boldsymbol{\omega}} = {}^I\boldsymbol{\omega}_m - \hat{\mathbf{b}}_g$ and ${}^G\hat{\mathbf{a}}_I = \mathbf{C}({}^I\hat{\mathbf{q}})^T({}^I\mathbf{a}_m - {}^I\hat{\mathbf{b}}) + {}^G\mathbf{g}$. The true states and their derivatives in Eq. 11 can be expressed in terms of the estimated states and error states according to their definitions. If this is then compared to Eq. 15, the continuous time error state kinematics can be derived. The reader is referred to Trawny³⁰ for the full derivation. The continuous time error-state kinematics are represented in matrix form as

$${}^G\dot{\tilde{\mathbf{x}}}_1 = \mathbf{F}_c {}^G\tilde{\mathbf{x}}_1 + \mathbf{G}_c \mathbf{n}, \quad (16)$$

where $\mathbf{n} = \left\{ \mathbf{n}_g^T \quad \mathbf{n}_{wg}^T \quad \mathbf{n}_a^T \quad \mathbf{n}_{wa}^T \right\}^T$ and the system matrix and noise input matrix

are given by the following:

$$\mathbf{F}_c = \begin{bmatrix} -[\mathbf{I}\hat{\boldsymbol{\omega}}\times] & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{C}(\mathbf{I}_G\hat{\mathbf{q}})^T[\mathbf{I}\hat{\mathbf{a}}\times] & \mathbf{0} & \mathbf{0} & -\mathbf{C}(\mathbf{I}_G\hat{\mathbf{q}})^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (17)$$

$$\mathbf{G}_c = \begin{bmatrix} -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{C}(\mathbf{I}_G\bar{\mathbf{q}})^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

The system noise is modeled by uncorrelated zero-mean white noise Gaussian processes with the following power spectral density matrix:

$$\mathbf{Q}_c = \begin{bmatrix} \sigma_g^2 \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_{wg}^2 \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_a^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{wa}^2 \mathbf{I} \end{bmatrix}, \quad (18)$$

where σ_g^2 , σ_{wg}^2 , σ_a^2 , and σ_{wa}^2 are the power spectral densities of the gyroscope noise, gyroscope bias random walk, accelerometer noise, and accelerometer bias random walk, respectively. For this application, the error-state dynamics are discretized by determining the state transition matrix Φ , and discrete-time error covariance matrix \mathbf{Q} using the following definitions³¹:

$${}_{i-1}^i \Phi = e^{\Delta t \mathbf{F}_c} \quad (19)$$

$${}_{i-1}^i \mathbf{Q} = \int_0^{\Delta t} e^{(\Delta t - \tau) \mathbf{F}_c} \mathbf{G}_c \mathbf{Q}_c \mathbf{G}_c^T e^{(\Delta t - \tau) \mathbf{F}_c^T} d\tau \quad (20)$$

where i is the current IMU sample time, and Δt is the IMU sample period. Equations 19 and 20 depend on the average accelerometer output, gyroscope output, and quaternion over the previous IMU time interval, and can be calculated either numerically³² or using the closed form solutions from citeRomuloutousTR. Typically, the IMU update rate is significantly faster than the camera frame-rate. If there are n IMU samples per imager sample, then the following recursive relationships can be

used to determine ${}^k_{k-1}\Phi$ and ${}^k_{k-1}\mathbf{Q}$, the total state transition matrix and covariance matrix between imager time $k-1$ and k :

$${}^{(k-1)n+i}_{(k-1)n}\Phi = {}^i_{i-1}\Phi {}^{(k-1)n+i-1}_{(k-1)n}\Phi, \quad (21)$$

$${}^{(k-1)n+i}_{(k-1)n}\mathbf{Q} = {}^i_{i-1}\Phi {}^{(k-1)n+i-1}_{(k-1)n}\mathbf{Q}^i_{i-1}\Phi^T + {}^i_{i-1}\mathbf{Q}. \quad (22)$$

At each new image time step k , the current previous IMU pose becomes the most recent pose in ${}^G\mathbf{x}_2$, and the oldest pose in ${}^G\mathbf{x}_2$ is discarded. This leads to the following linear mapping from the previous to the current error states:

$$\begin{Bmatrix} {}^G\tilde{\mathbf{x}}_{1,k} \\ {}^G\tilde{\mathbf{x}}_{2,k} \end{Bmatrix} = \begin{bmatrix} {}^k_{k-1}\Phi & \mathbf{0}_{15 \times 6l} \\ \mathbf{L} & \mathbf{M} \end{bmatrix} \begin{Bmatrix} {}^G\tilde{\mathbf{x}}_{1,k-1} \\ {}^G\tilde{\mathbf{x}}_{2,k-1} \end{Bmatrix}, \quad (23)$$

$$\mathbf{L} = \begin{bmatrix} \mathbf{N} \\ \mathbf{0}_{6(N-1) \times 18} \end{bmatrix}, \quad (24)$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{0}_{6 \times 6(l-1)} & \mathbf{0}_{6 \times 6} \\ \mathbf{I}_{6(N-1) \times 6(l-1)} & \mathbf{0}_{6(l-1) \times 6} \end{bmatrix}, \quad (25)$$

$$\mathbf{N} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}. \quad (26)$$

This mapping is then used to propagate the error-state covariance matrix

$$\mathbf{P}_{k|k-1} = \begin{bmatrix} {}^k_{k-1}\Phi & \mathbf{0}_{15 \times 6l} \\ \mathbf{L} & \mathbf{M} \end{bmatrix} \mathbf{P}_{k-1|k-1} \begin{bmatrix} {}^k_{k-1}\Phi & \mathbf{0}_{15 \times 6l} \\ \mathbf{L} & \mathbf{M} \end{bmatrix}^T + \begin{bmatrix} {}^k_{k-1}\mathbf{Q} & \mathbf{0}_{15 \times 6l} \\ \mathbf{0}_{6l \times 15} & \mathbf{0}_{6l \times 6l} \end{bmatrix}. \quad (27)$$

4. OSC-EKF Update

The OSC-EKF utilizes the estimates and Fisher information from a user-defined SAM problem in the EKF update step. The main assumption is that the SAM solution provides motion constraints between the image frames relative to some local frame. The measurements used in the SAM problem are assumed to have an output of the form

$$\mathbf{z} = h({}^L\mathbf{x}, \boldsymbol{\eta}) + \mathbf{v}, \quad (28)$$

where h is a vector-valued function of the IMU poses relative to the local frame ${}^L\mathbf{x}$ and a collection of nuisance parameters $\boldsymbol{\eta}$. The measurements are corrupted with random noise with assumed distribution $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_v)$.

In the example provided in this work, the measurements are pixel coordinates of feature point correspondences tracked by a pair of calibrated cameras. The nuisance parameters are the 3-D locations of those feature points. However, the nuisance parameter vector could easily be extended to include camera calibration errors. In a different application, other sensors could be included such as odometers, LIDAR sensors, or baro-altimeters, and the nuisance parameter vector extended to include deterministic errors in those sensors. It is important however that the IMU output contribute no information to the SAM information matrix, because this information is already being used by the EKF via the propagation model. The IMU poses relative to the local frame are defined as

$${}^L\mathbf{x} = \left\{ {}^L\bar{\mathbf{q}}^T \quad {}^L\mathbf{p}_{I_k}^T \quad \dots \quad {}^L\bar{\mathbf{q}}^T \quad {}^L\mathbf{p}_{I_{k-(l-1)}}^T \right\}^T, \quad (29)$$

$$\mathbf{C} \left({}^L\bar{\mathbf{q}} \right) = \mathbf{C} \left({}^L\bar{\mathbf{q}} \right) \mathbf{C} \left({}^L\bar{\mathbf{q}} \right)^T, \quad (30)$$

$${}^L\mathbf{p}_{I_k} = \mathbf{C} \left({}^L\bar{\mathbf{q}} \right) \left({}^G\mathbf{p}_{I_k} - {}^G\mathbf{p}_L \right). \quad (31)$$

This definition is particular to the convention of using the reference frame defined by the oldest IMU pose in the sliding window as the local frame. Notice that if there are l poses in the sliding window, only $l - 1$ of them will be independent in the SAM motion with relative constraints.

The OSC-EKF update is performed when the sliding window has progressed l image samples beyond the previous update. This is to ensure that the measurements used in the EKF update are uncorrelated with previous measurements, and no information is re-used. The SAM problem computes the maximum likelihood estimate (MLE) of ${}^L\mathbf{x}$, which will be denoted ${}^L\hat{\mathbf{x}}_{MLE}$, and the MLE of the nuisance parameters $\hat{\boldsymbol{\eta}}_{MLE}$. The predicted local states ${}^L\hat{\mathbf{x}}$ can be computed by substituting ${}^G\hat{\mathbf{x}}$ for ${}^G\mathbf{x}$ in Eqns. 30 and 31 and this prediction makes a good initial guess for ${}^L\hat{\mathbf{x}}_{MLE}$. That is

$$\mathbf{C} \left({}^L\hat{\bar{\mathbf{q}}} \right) \equiv \mathbf{C} \left({}^L\hat{\bar{\mathbf{q}}} \right) \mathbf{C} \left({}^L\hat{\bar{\mathbf{q}}} \right)^T, \quad (32)$$

$${}^L\hat{\mathbf{p}}_{I_k} \equiv \mathbf{C} \left({}^L\hat{\bar{\mathbf{q}}} \right) \left({}^G\hat{\mathbf{p}}_{I_k} - {}^G\hat{\mathbf{p}}_L \right). \quad (33)$$

Assuming the measurement errors in Eq. 28 are independent, the MLE is equivalent

to the following nonlinear least squares problem:

$$\left\{ {}^L\hat{\mathbf{x}}_{MLE} \quad , \quad \hat{\boldsymbol{\eta}}_{MLE} \right\} = \underset{{}^L\mathbf{x}, \boldsymbol{\eta}}{\operatorname{argmin}} \left(\mathbf{z} - h \left({}^L\mathbf{x}, \boldsymbol{\eta} \right) \right)^T \mathbf{R}_v^{-1} \left(\mathbf{z} - h \left({}^L\mathbf{x}, \boldsymbol{\eta} \right) \right). \quad (34)$$

The error-state vector relating the predicted local states to the true local states is given by

$${}^L\tilde{\mathbf{x}} = \left\{ {}^{I_k}\tilde{\boldsymbol{\phi}}^T \quad {}^L\tilde{\mathbf{p}}_{I_k}^T \quad \dots \quad {}^{I_{k-(l-1)}}\tilde{\boldsymbol{\phi}}^T \quad {}^L\tilde{\mathbf{p}}_{I_{k-(l-1)}}^T \right\}^T, \quad (35)$$

$${}^{I_k}_L\tilde{\mathbf{q}} = {}^{I_k}_L\tilde{\mathbf{q}} \otimes {}^{I_k}_L\hat{\mathbf{q}}, \quad (36)$$

$${}^{I_k}_L\tilde{\mathbf{q}} \approx \begin{Bmatrix} {}^{I_k}\tilde{\boldsymbol{\phi}}_{\frac{1}{2}} \\ 1 \end{Bmatrix}, \quad (37)$$

$${}^L\mathbf{p}_{I_k} = {}^L\hat{\mathbf{p}}_{I_k} + {}^L\tilde{\mathbf{p}}_{I_k}. \quad (38)$$

The error states relating the MLE local states to the true local states are denoted ${}^L\tilde{\mathbf{x}}_{MLE}$, and have the same form. The problem in Eq. 34 is usually solved using a Gauss–Newton or Levenberg–Marquardt iterative approach. The Jacobian of the measurement equation with respect to the error-states takes on the form

$$\mathbf{J}_{MLE} = \begin{bmatrix} \mathbf{J}_{{}^L\tilde{\mathbf{x}}_{MLE}} & , & \mathbf{J}_{{}^L\tilde{\boldsymbol{\eta}}_{MLE}} \end{bmatrix}. \quad (39)$$

The information matrix at convergence is approximated by

$$\mathbf{H}_{MLE} \approx \mathbf{J}_{MLE}^T \mathbf{R}_v^{-1} \mathbf{J}_{MLE} \equiv \begin{bmatrix} \mathbf{H}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} & \mathbf{H}_{\tilde{\mathbf{x}}\tilde{\boldsymbol{\eta}}} \\ \mathbf{H}_{\tilde{\boldsymbol{\eta}}\tilde{\mathbf{x}}} & \mathbf{H}_{\tilde{\boldsymbol{\eta}}\tilde{\boldsymbol{\eta}}} \end{bmatrix}. \quad (40)$$

The MLE is asymptotically efficient, so if there are a sufficient number of measurements it is reasonable to make the following assumptions:

$$E \left[{}^L\tilde{\mathbf{x}}_{MLE} \right] \approx \mathbf{0}, \quad (41)$$

$$E \left[{}^L\tilde{\mathbf{x}}_{MLE} {}^L\tilde{\mathbf{x}}_{MLE}^T \right] \approx \left(\mathbf{H}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} - \mathbf{H}_{\tilde{\mathbf{x}}\tilde{\boldsymbol{\eta}}} \mathbf{H}_{\tilde{\boldsymbol{\eta}}\tilde{\boldsymbol{\eta}}}^{-1} \mathbf{H}_{\tilde{\boldsymbol{\eta}}\tilde{\mathbf{x}}} \right)^{-1}. \quad (42)$$

Because the MLE is also asymptotically normal, it will be assumed ${}^L\tilde{\mathbf{x}}_{MLE} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{MLE})$, where \mathbf{R}_{MLE} is given by Eq. 42. The updated ${}^L\hat{\mathbf{x}}_{MLE}$ can therefore be treated as a direct measurement of ${}^L\mathbf{x}$ corrupted by zero-mean Gaussian noise. The pose residuals between ${}^L\hat{\mathbf{x}}_{MLE}$ and ${}^L\hat{\mathbf{x}}$ are formulated in terms of the

error states

$${}^{I_k} \tilde{\mathbf{q}} = {}^{I_k} \hat{\mathbf{q}}_{MLE} \otimes {}^{I_k} \hat{\mathbf{q}}^{-1}, \quad (43)$$

$${}^L \tilde{\mathbf{p}}_{I_k} = {}^L \hat{\mathbf{p}}_{I_k MLE} - {}^L \hat{\mathbf{p}}_{I_k}. \quad (44)$$

The small rotation angles ${}^{I_k} \tilde{\boldsymbol{\phi}}$ are computed from Eq. 37. The EKF update requires a linear transformation that maps the tracked error-states to the measurement residuals. The tracked error states are ${}^G \tilde{\mathbf{x}}$, and the measurement residuals are ${}^L \tilde{\mathbf{x}}$, so the linear transformation can be found by applying perturbation analysis to Eqs. 30 and 31. It is helpful to denote separately the global error states for the local frame by position error ${}^G \tilde{\mathbf{p}}_L$ and small angles ${}^L \tilde{\boldsymbol{\psi}}$. Writing the true states in terms of the estimated and error states turns Eq. 30 into

$$\left(\mathbf{I} - \left[{}^{I_k} \tilde{\boldsymbol{\phi}} \times \right] \right) \mathbf{C} \left({}^{I_k} \hat{\mathbf{q}} \right) \approx \left(\mathbf{I} - \left[{}^{I_k} \tilde{\boldsymbol{\theta}} \times \right] \right) \mathbf{C} \left({}^{I_k} \hat{\mathbf{q}} \right) \left[\left(\mathbf{I} - \left[{}^L \tilde{\boldsymbol{\psi}} \times \right] \right) \mathbf{C} \left({}^L \hat{\mathbf{q}} \right) \right]^T. \quad (45)$$

By applying properties of the skew-symmetric matrix, and using definition from Eq. 32, this simplifies to

$$\left[{}^{I_k} \tilde{\boldsymbol{\phi}} \times \right] \mathbf{C} \left({}^{I_k} \hat{\mathbf{q}} \right) \approx \left[{}^{I_k} \tilde{\boldsymbol{\theta}} \times \right] \mathbf{C} \left({}^{I_k} \hat{\mathbf{q}} \right) - \left[\mathbf{C} \left({}^{I_k} \hat{\mathbf{q}} \right) {}^L \tilde{\boldsymbol{\psi}} \times \right] \mathbf{C} \left({}^{I_k} \hat{\mathbf{q}} \right). \quad (46)$$

So the linear transformation for the small angles is

$${}^{I_k} \tilde{\boldsymbol{\phi}} \approx \left[\mathbf{I} \quad -\mathbf{C} \left({}^{I_k} \hat{\mathbf{q}} \right) \right] \begin{Bmatrix} {}^{I_k} \tilde{\boldsymbol{\theta}} \\ {}^L \tilde{\boldsymbol{\psi}} \end{Bmatrix}. \quad (47)$$

In a similar fashion, Eq. 31 becomes

$${}^L \hat{\mathbf{p}}_{I_k} + {}^L \tilde{\mathbf{p}}_{I_k} = \left(\mathbf{I} - \left[{}^L \tilde{\boldsymbol{\psi}} \times \right] \right) \mathbf{C} \left({}^L \hat{\mathbf{q}} \right) \left({}^G \hat{\mathbf{p}}_{I_k} + {}^G \tilde{\mathbf{p}}_{I_k} - {}^G \hat{\mathbf{p}}_L - {}^G \tilde{\mathbf{p}}_L \right). \quad (48)$$

By using the definition from Eq. 33, and neglecting products of error-states, Eq. 48 simplifies to

$${}^L \tilde{\mathbf{p}}_{I_k} \approx \left[\mathbf{C} \left({}^L \hat{\mathbf{q}} \right) \quad \left[{}^L \hat{\mathbf{p}}_I \times \right] \quad -\mathbf{C} \left({}^L \hat{\mathbf{q}} \right) \right] \begin{Bmatrix} {}^G \tilde{\mathbf{p}}_I \\ {}^L \tilde{\boldsymbol{\psi}} \\ {}^G \tilde{\mathbf{p}}_L \end{Bmatrix}. \quad (49)$$

Through proper arranging of Eqs. 47 and 49, the total transformation used in the

EKF between ${}^G\tilde{\mathbf{x}}$ and ${}^L\tilde{\mathbf{x}}$ is defined by

$${}^L\tilde{\mathbf{x}} = \mathbf{H}^G\tilde{\mathbf{x}}. \quad (50)$$

The OSC-EKF update estimates the current error states through the standard Kalman update. The residual covariance is given by

$$\mathbf{S} = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_{MLE}. \quad (51)$$

The Kalman gain is calculated as

$$\mathbf{K} = \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}^{-1}. \quad (52)$$

The error states are calculated

$${}^G\tilde{\mathbf{x}} = \mathbf{K}{}^L\tilde{\mathbf{x}}. \quad (53)$$

The error-state covariance is updated by

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{R}_{MLE}\mathbf{K}^T. \quad (54)$$

The error states are then used to correct the estimated states using Eqs. 4 and 5, and subsequently set to zero.

5. Practical MLE Issues

One obvious potential flaw in OSC-EKF is that without sufficient information, the MLE can fail to converge or converge to a local minimum. For example, in the stereo vision case this can occur if too few features are extracted and matched between cameras. In the case of monocular vision, a lack of motion will cause the Hessian matrix in Eq. 40 to become ill-conditioned or singular. Although optimization methods such as the Levenberg Marquardt algorithm can compensate for this and prevent divergence, a poorly conditioned problem is less likely to progress towards a well-defined global minimum. The OSC-EKF is not well suited to acquiring images such that they will have strong geometry.¹⁴ While the window of poses could be updated with this goal in mind (instead of at fixed time intervals), the ability to do this is limited because over time the small perturbation assumptions used in Section 3 will become less valid. It is important that the predicted IMU trajectory not contribute to the Fisher information of the MLE, because this would violate

the EKF assumption that the state errors and measurement errors are independent. However, it has been found that using the predicted states and covariances as priors in the MLE estimate can significantly aid in the convergence process. Once stable convergence has been reached, the Fisher information matrix can be calculated without priors for use in the EKF update. While this method may introduce a small bias in the MLE, it was found to significantly improve the overall performance of the algorithm, particularly when using monocular vision.

6. Experimental Results

The OSC-EKF as described was implemented in MATLAB³³ and evaluated with the EuRoC micro aerial vehicle (MAV) datasets.²⁹ The datasets were created by flying a AscTex Firefly MAV equipped with 2 global shutter greyscale cameras and an ADIS16448 IMU through 2 separate indoor environments. Position and attitude ground truth measurements are provided with the datasets from a postprocessing solution aided by a VICON system or a Leica laser tracker depending on the environment. The IMU data was provided at 200 Hz, and the image data at 20 Hz. Camera and IMU calibrations were provided, so only IMU biases were included in the state vector as described in Section 2. IMU sensor noise parameters were provided with the dataset, but they were not used. Instead, the discrete state error covariance matrix was estimated from the IMU data and ground truth and the modeled power spectral densities adjusted so that Eq. 20 produced a similar matrix. This noise model produced more consistent estimates when incorporated into the OSC-EKF.

A rudimentary image-processing front end was created. Images were first histogram-equalized to aid in tracking features in low-light images. Scale invariant feature transform (SIFT) feature points were tracked using the VLFeat³⁴ MATLAB implementation. Outliers were rejected between frames using RANSAC, and between cameras using epipolar constraints. The OSC-EKF used a window size of 10 frames. The SAM problem was constructed using an inverse-depth feature position parameterization as described in Appendix A. In the stereo implementation, the Hessian was naturally well conditioned because only features seen in both cameras were included in the optimization. The SAM was solved with a custom Levenberg Marquardt solver. In the monocular implementation, the convergence was aided by the state prediction as described in Section 5. In both cases, residual outliers were re-

jected by keeping those points with a residual norm less than 6 times the median residual norm.

The algorithm performance from 2 of the dataset trajectories is presented here. The “MH 05 difficult” sequence contained rapid motion of the MAV in a dark machine hall. The “V2 03 difficult” sequence contained rapid motion of the MAV and motion blur in the images. The ground truth and stereo OSC-EKF estimated trajectories of these sequences are shown in Fig. 2. The actual position errors are displayed for both the stereo and monocular implementations in Figs. 3 and 4. The errors were all less than 1 percent of the distance traveled except for the monocular implementation on the V2 03 difficult sequence. The monocular implementations perform slightly worse than the stereo implementations, which is to be expected because no absolute scale information can be extracted from the vision measurements.

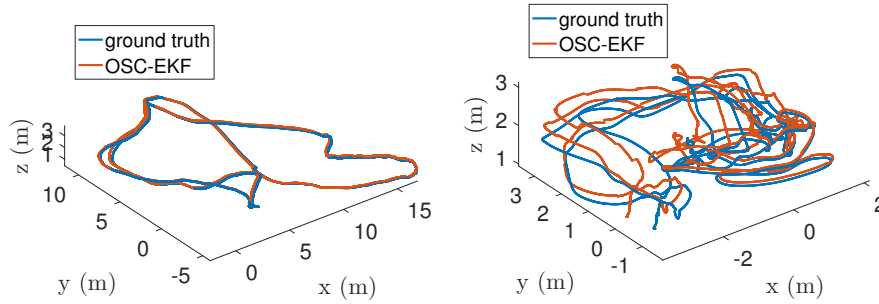


Fig. 2 Stereo position tracking (MH 05 difficult left, V2 03 difficult right)

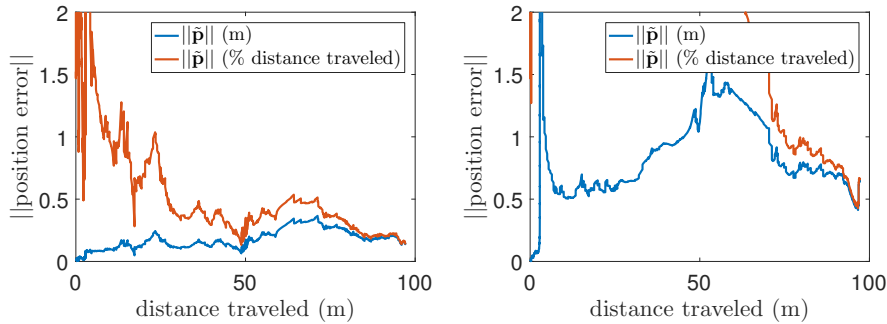


Fig. 3 Position error for the MH 05 difficult sequence (stereo left, mono right)

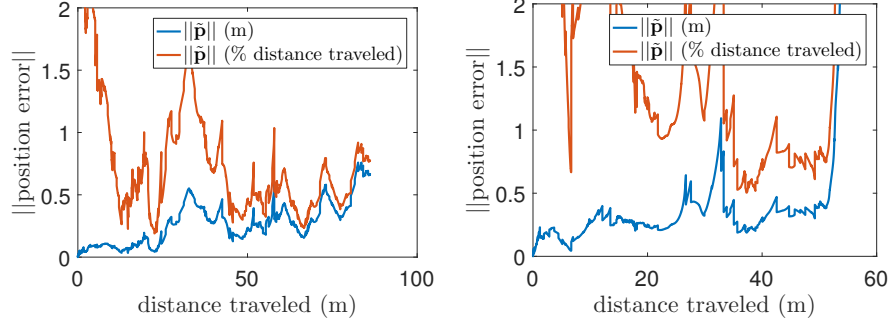


Fig. 4 Position error for the V2 03 difficult sequence (stereo left, mono right)

The consistency was analyzed by examining the ratio of the norm of the position error to the square root of the sum of the position error covariance terms reported by the OSC-EKF. Ideally, this ratio should fluctuate around 0.65, and this seems to be the case for both the stereo and monocular implementations on the MH 05 difficult sequence as illustrated in Fig. 5. Both implementations were less consistent on the V2 03 difficult sequence as illustrated in Fig. 6. It is likely that much of the blame for this is due to updates in which very few feature points were successfully tracked for use in the SAM problem. The number of features used in each OSC-EKF update is labeled on the right axis of Figs. 5 and 6. Some updates in the V2 03 difficult sequence have fewer than 30 useable feature points with which to solve a 10-frame SAM problem. While the algorithm does not diverge, it is likely that the asymptotic normality assumption is violated and the \mathbf{R}_{MLE} used to update the OSC-EKF is inaccurate. Overall, the similarity in consistency between stereo and monocular implementations demonstrates the generality of the OSC-EKF algorithm.

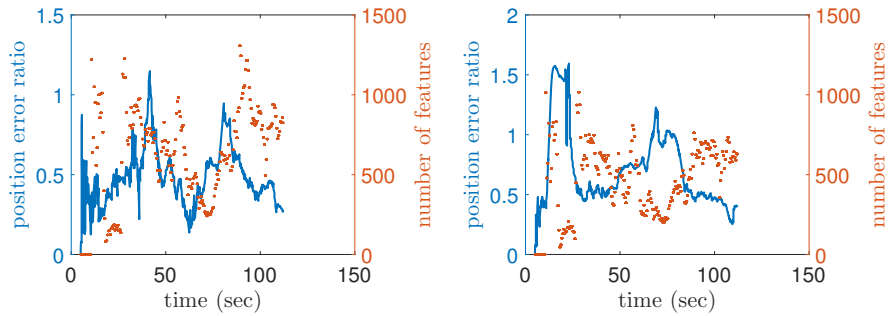


Fig. 5 Consistency analysis for the MH 05 difficult sequence (stereo left, mono right)

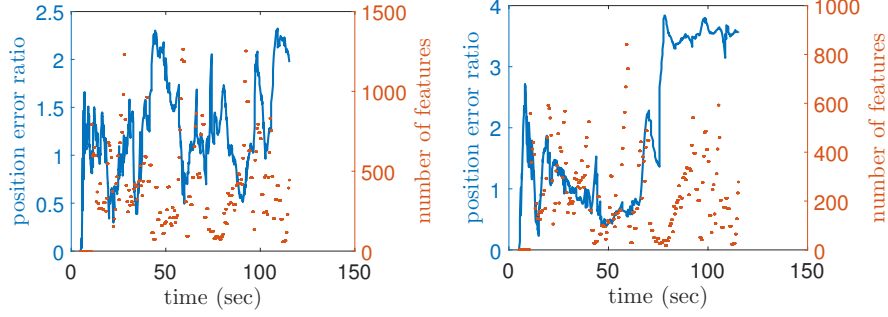


Fig. 6 Consistency analysis for the V2 03 difficult sequence (stereo left, mono right)

7. Conclusions and Future Work

A generalized version of the OSC-EKF has been presented that can use any combination of sensors to create the relative pose constraints used in the update step. The implementation is efficient because Kalman filtering is naturally recursive, and the state size is fixed and user definable. The generality of the algorithm was demonstrated by implementing both a stereo and monocular SAM construct with which to generate the relative pose constraints without changing the structure of the OSC-EKF itself. Reasonable drift rates and consistency were demonstrated on a challenging MAV dataset. The navigation performance was mainly hindered by the success of the visual feature tracker, which was not the main objective of this work.

Reliable implementation will require a more robust visual front-end for tracking features. Efficient execution will require a method of dealing with processing delays. The OSC-EKF is actually well suited to this because allowing for delayed EKF updates would enable parallel processing of the OSC-EKF propagation and the previous SAM problem. Extended experimentation with different sensors, and implementing loop closure could also serve as future research directions.

8. References

1. Shi J, Tomasi C. Good features to track. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition; 1994 Jun 21–23; Seattle, WA. Los Alamitos (CA): IEEE Computer Society; 1994. p. 593–600.
2. Tuytelaars T, Mikolajczyk K. Local invariant feature detectors: a survey. *Found Trends Comput Graph*. 2008;3(3):177–280.
3. Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry. In: Proc. of the IEEE International Conference on Robotics and Automation; 2014 May 31–Jun 7; Hong Kong, China. New York (NY): IEEE; 2014.
4. Google. Google Project Tango. [accessed 2016 Oct 19]; <https://www.google.com/atap/projecttango>.
5. Durrant-Whyte H, Majumder S, Thrun S, de Battista M, Scheduling S. A bayesian algorithm for simultaneous localisation and map building. In: Jarvis RA, Zelinsky A, editors. *Robotics research: the tenth international symposium*; Berlin (Germany): Springer Berlin Heidelberg; 2003. p. 49–60.
6. Kim J, Sukkarieh S. Real-time implementation of airborne inertial-SLAM. *Robotics and Autonomous Systems*. 2007;55(1):62–71.
7. Li M, Kim B, Mourikis AI. Real-time motion estimation on a cellphone using inertial sensing and a rolling-shutter camera. In: Proc. of the IEEE International Conference on Robotics and Automation; 2013 May 6–10; Karlsruhe, Germany. New York (NY): IEEE; 2013. p. 4697–4704.
8. Loianno G, Cross G, Qu C, Mulgaonkar Y, Hesch JA, Kumar V. Flying smartphones: automated flight enabled by consumer electronics. *IEEE Robotics Automation Magazine*. 2015;22(2):24–32.
9. Weiss S, Siegwart R. Real-time metric state estimation for modular vision-inertial systems. In: Proc. of the IEEE International Conference on Robotics and Automation; 2011 May 9–13; Shanghai, China. New York (NY): IEEE; 2011. p. 4531–4537.

10. Williams S, Indelman V, Kaess M, Roberts R, Leonard JJ, Dellaert F. Concurrent filtering and smoothing: a parallel architecture for real-time navigation and full smoothing. *International Journal of Robotics Research*. 2014;33(12):1544–1568.
11. Kerl C, Sturm J, Cremers D. Robust odometry estimation for rgb-d cameras. In: *Proc. of the IEEE International Conference on Robotics and Automation*; 2013 May 6–10; Karlsruhe, Germany. New York (NY): IEEE; 2013.
12. Scaramuzza D, Fraundorfer F. Visual odometry [tutorial]. *IEEE Robotics Automation Magazine*. 2011;18(4):80–92.
13. Fraundorfer F, Scaramuzza D. Visual odometry: Part ii: matching, robustness, optimization, and applications. *IEEE Robotics Automation Magazine*. 2012;19(2):78–90.
14. Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW. Bundle adjustment — a modern synthesis. In: Triggs B, Zisserman A, Szeliski R, editors. *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms. Proceedings*; 2000 Sep 21–22; Corfu, Greece. Berlin (Germany): Springer Berlin Heidelberg; 2000. p. 298–372.
15. Hartley R, Zisserman A. *Multiple view geometry in computer vision*. New York (NY): Cambridge University Press; 2004.
16. Leutenegger S, Lynen S, Bosse M, Siegwart R, Furgale P. Keyframe-based visual-inertial odometry using nonlinear optimization. *International Journal of Robotics Research*. 2015;34(3):314–334.
17. Kaess M, Ranganathan A, Dellaert F. iSAM: incremental smoothing and mapping. *IEEE Transactions on Robotics*. 2008;24(6):1365–1378.
18. Kaess M, Johannsson H, Roberts R, Ila V, Leonard J, Dellaert F. iSAM2: incremental smoothing and mapping using the Bayes tree. *International Journal of Robotics Research*. 2012;31:217–236.
19. Indelman V, Dellaert F. Incremental light bundle adjustment: probabilistic analysis and application to robotic navigation. In: Sun Y, Behal A, Chung CKR, editors. *New development in robot vision*; Berlin (Germany): Springer Berlin Heidelberg; 2015. p. 111–136.

20. Forster C, Carlone L, Dellaert F, Scaramuzza D. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In: *Proceedings of Robotics: Science and Systems*; 2015 Jul 13–27; Rome, Italy.
21. Lupton T, Sukkarieh S. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*. 2012;28(1):61–76.
22. Mourikis AI, Roumeliotis SI. A multi-state constraint Kalman filter for vision-aided inertial navigation. In: *Proceedings of the IEEE International Conference on Robotics and Automation*; 2007 Apr 10–14; Rome, Italy. New York (NY): IEEE; 2007. p. 3565–3572.
23. Roumeliotis SI, Burdick JW. Stochastic cloning: a generalized framework for processing relative state measurements. In: *Proceedings of the IEEE International Conference on Robotics and Automation*; 2002 May 11–15; Washington, DC. New York (NY): IEEE; 2002. p. 1788–1795.
24. Huang G, Kaess M, Leonard J. Towards consistent visual-inertial navigation. In: *Proc. of the IEEE International Conference on Robotics and Automation*; 2014 May 31–Jun 7; Hong Kong, China. New York (NY): IEEE; 2014. p. 4926–4933.
25. Huang G, Trawny N, Mourikis AI, Roumeliotis SI. On the consistency of multi-robot cooperative localization. In: *Proc. of the Robotics: Science and Systems*; 2009 Jun 31–Jul 1; Seattle, WA. Cambridge, MA: MIT Press; 2009. p. 65–72.
26. Hesch JA, Kottas DG, Bowman SL, Roumeliotis SI. Towards consistent vision-aided inertial navigation. In: Frazzoli E, Lozano-Perez T, Roy N, Rus D, editors. *Algorithmic foundations of robotics X: proceedings of the tenth workshop on the algorithmic foundations of robotics*; Berlin (Germany): Springer Berlin Heidelberg; 2013. p. 559–574.
27. Li M, Mourikis A. High-precision, consistent EKF-based visual-inertial odometry. *International Journal of Robotics Research*. 2013;32(6):690–711.
28. Huang G, Eickenhoff K, Leonard J. Optimal-state-constraint EKF for visual-inertial navigation. In: *Proc. of the International Symposium on Robotics Research*; 2015 Sep12–15; Sestri Levante, Italy.

29. Burri M, Nikolic J, Gohl P, Schneider T, Rehder J, Omari S, Achtelik MW, Siegwart R. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*. 2016;35(10):1157–1163.
30. Trawny N, Roumeliotis SI. Indirect Kalman filter for 3D attitude estimation. Minneapolis (MN): University of Minnesota, Dept. of Comp. Sci. & Eng.; 2005 Mar. Report No.: 2005–002, Rev. 57.
31. Simon D. Optimal state estimation. Hoboken (NJ): John Wiley and Sons, Inc.; 2006.
32. Van Loan C. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*. 1978;23(3):395–404.
33. MATLAB Release 2016a. Natick (MA): The MathWorks, Inc.; 2016.
34. Vedaldi A, Fulkerson B. VLFeat: An open and portable library of computer vision algorithms. VLFeat.org; 2008 [accessed 2016 Apr-20]. <http://www.vlfeat.org/>.

Appendix A. Feature Parameterization

The example structure and motion (SAM) problem uses calibrated cameras to determine the maximum likelihood estimate (MLE) the local error states ${}^L\hat{\mathbf{x}}_{MLE}$ and nuisance parameters $\hat{\boldsymbol{\eta}}_{MLE}$ (feature point locations). A computer-vision front-end tracks visual feature points between consecutive frames and between cameras. The measurements used are the direct pixel coordinates of these points from each camera; no prior rectification or depth estimation is performed by the front-end. In parameterizing these measurements, it is useful to start with the position of feature point f relative to the reference frame attached to the camera j at image time-step k , denoted C_{jk}

$$C_{jk}\mathbf{p}_f = \mathbf{C}\left(\begin{smallmatrix} C_{jk} \\ I_k \end{smallmatrix}\bar{\mathbf{q}}\right) \left(\mathbf{C}\left(\begin{smallmatrix} I_k \\ L \end{smallmatrix}\bar{\mathbf{q}}\right) ({}^L\mathbf{p}_f - {}^L\mathbf{p}_{I_k}) - {}^{I_k}\mathbf{p}_{C_{jk}} \right). \quad (\text{A-1})$$

The position and rotation of the camera relative to the inertial measurement unit ${}^{I_k}\mathbf{p}_{C_{jk}}$, and $\begin{smallmatrix} C_{jk} \\ I_k \end{smallmatrix}\bar{\mathbf{q}}$, are assumed to be known from the calibration and constant. The feature points use an inverse-depth parameterization [citeInverseDepth](#)

$${}^L\mathbf{p}_f = \frac{1}{\rho_f} \begin{Bmatrix} \alpha_f \\ \beta_f \\ 1 \end{Bmatrix}. \quad (\text{A-2})$$

The feature point in the left camera frame then becomes

$$C_{jk}\mathbf{p}_f = \frac{1}{\rho_f} \left[\mathbf{C}\left(\begin{smallmatrix} C_{jk} \\ I_k \end{smallmatrix}\bar{\mathbf{q}}\right) \left(\mathbf{C}\left(\begin{smallmatrix} I_k \\ L \end{smallmatrix}\bar{\mathbf{q}}\right) \left(\begin{Bmatrix} \alpha_f \\ \beta_f \\ 1 \end{Bmatrix} - \rho_f {}^L\mathbf{p}_{I_k} \right) - \rho_f {}^{I_k}\mathbf{p}_{C_{jk}} \right) \right]. \quad (\text{A-3})$$

The term in the brackets can be replaced by the nonlinear functions h_x, h_y, h_z

$$C_{jk}\mathbf{p}_f = \frac{1}{\rho_f} \begin{bmatrix} h_x \left(\alpha_f, \beta_f, \rho_f, \begin{smallmatrix} I_k \\ L \end{smallmatrix}\bar{\mathbf{q}}, {}^L\mathbf{p}_{I_k} \right) \\ h_y \left(\alpha_f, \beta_f, \rho_f, \begin{smallmatrix} I_k \\ L \end{smallmatrix}\bar{\mathbf{q}}, {}^L\mathbf{p}_{I_k} \right) \\ h_z \left(\alpha_f, \beta_f, \rho_f, \begin{smallmatrix} I_k \\ L \end{smallmatrix}\bar{\mathbf{q}}, {}^L\mathbf{p}_{I_k} \right) \end{bmatrix}. \quad (\text{A-4})$$

The normalized pixel coordinates $\{u, v\}$, are then calculated as

$$\begin{Bmatrix} u \\ v \end{Bmatrix}_{fjk} = \begin{Bmatrix} \frac{h_x(\alpha_f, \beta_f, \rho_f, {}^{I_k} \bar{\mathbf{q}}, {}^L \mathbf{p}_{I_k})}{h_z(\alpha_f, \beta_f, \rho_f, {}^{I_k} \bar{\mathbf{q}}, {}^L \mathbf{p}_{I_k})} \\ \frac{h_y(\alpha_f, \beta_f, \rho_f, {}^{I_k} \bar{\mathbf{q}}, {}^L \mathbf{p}_{I_k})}{h_z(\alpha_f, \beta_f, \rho_f, {}^{I_k} \bar{\mathbf{q}}, {}^L \mathbf{p}_{I_k})} \end{Bmatrix}. \quad (\text{A-5})$$

Equation A-5 is the general form used to calculate normalized pixel coordinates for every feature point observed in the SAM window in both cameras.

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional.
BA	bundle adjustment.
EKF	extended Kalman filter.
GPS	global positioning system.
IMU	inertial measurement unit.
LIDAR	light detection and ranging.
MAV	micro aerial vehicle.
MLE	maximum likelihood estimate.
MSCKF	multiple state constraint filter.
OSC	optimal state constraint.
SAM	structure and motion.
SLAM	simultaneous localization and mapping.
SWaP	size, weight, and power.
VINS	visual inertial navigation system.
VIO	visual inertial odometry.

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO L
IMAL HRA MAIL & RECORDS MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 DREXEL UNIVERSITY
(PDF) B C CHANG

2 GEORGIA INSTITUTE OF TECHNOL-
(PDF) OGY
M COSTELLO
J ROGERS

1 UNIVERSITY OF DELAWARE
(PDF) G HUANG

7 RDECOM ARDEC
(PDF) J CHOI
R DECKER
M GUEVARA
P MAGNOTTI
T RECCHIA
C STOUT
W TOLEDO

4 NSWCDD
(PDF) L STEELMAN
K PAMADI
H MALIN
J FRAYSSE

2 RDECOM AMRDEC
(PDF) B GRANTHAM
K RISKO

3 AFRL
(PDF) T KAUSUTIS
A RUTKOWSKI
K BRINK

1 RDECOM ARL
(PDF) RDRL-SER-L
J CONROY

ABERDEEN PROVING GROUND

19 DIR USARL
(PDF) RDRL WML A
L STROHM
RDRL WML E
I CELMINS
L FAIRFAX
F FRESCONI
J VASILE
RDRL WML F
J MALEY
M ILG
M HAMAOU
M DON
B DAVIS
B ALLIK
C MILLER
J HALLAMEYER
D EVERSON
B KLINE
B NELSON
B TOPPER
G BROWN
RDRL WMP
D LYON